Programmation Android
I. Introduction

Plan

Plan

Application mobile

Caractéristiques

- Cible : **appareils électroniques mobiles autonomes** (smartphone, tablette, etc.)

- ⇒ **Contexte logiciel hétérogène** :
  - **non connecté** : pas d'accès réseau / connectivité limitée…
  - **connecté** : LAN / WAN …
  - **localisé (ou non)** : **GPS**, géolocalisation IP !
  - …

- ⇒ **Contexte matériel hétérogène** :
  - **ressources plus faibles** : CPU / RAM / DD / ROM
  - ⇒ la **consommation** des ressources doit être **minimale**
  - ⇒ **Résolutions d'affichage hétéroclites** ⇒ **GUI adaptative**
  - …

Application mobile

Principaux OS

- **Android** (Google) : Linux, version 12 (octobre 2021)

- **iOS** (Apple) noyau hybride XNU : Mach(libre) + BSD (Unix)

- **Tizen** (*Linux Foundation*), supporté par Samsung (montres connectées)

- **Librem 5** (*Purism, PureOS*), vie privée / sécurité

- **Windows 10 mobile** (Microsoft) (arrêté en octobre 2017)

- **Ubuntu Touch** (Canonical) : Linux (arrêté en avril 2017)

- **BlackBerry** OS, **QNX** : Unix (arrêté en 2015)

- **Firefox OS** (Mozilla Corporation) : Linux (arrêté en 2015)

- ⟨ ▸ plus d'information ⟩

Plan

Android

Origine

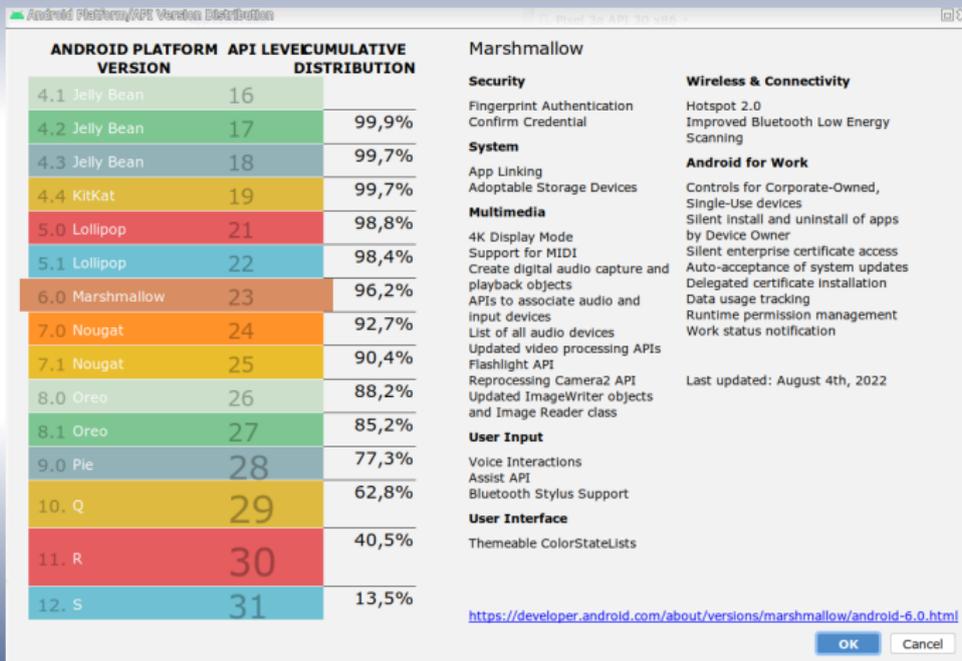- Initialement un OS pour appareil photo

- Acheté par Google en 2005

- ▸ Wikipédia

Diffusion

Google I/O, mai 2019

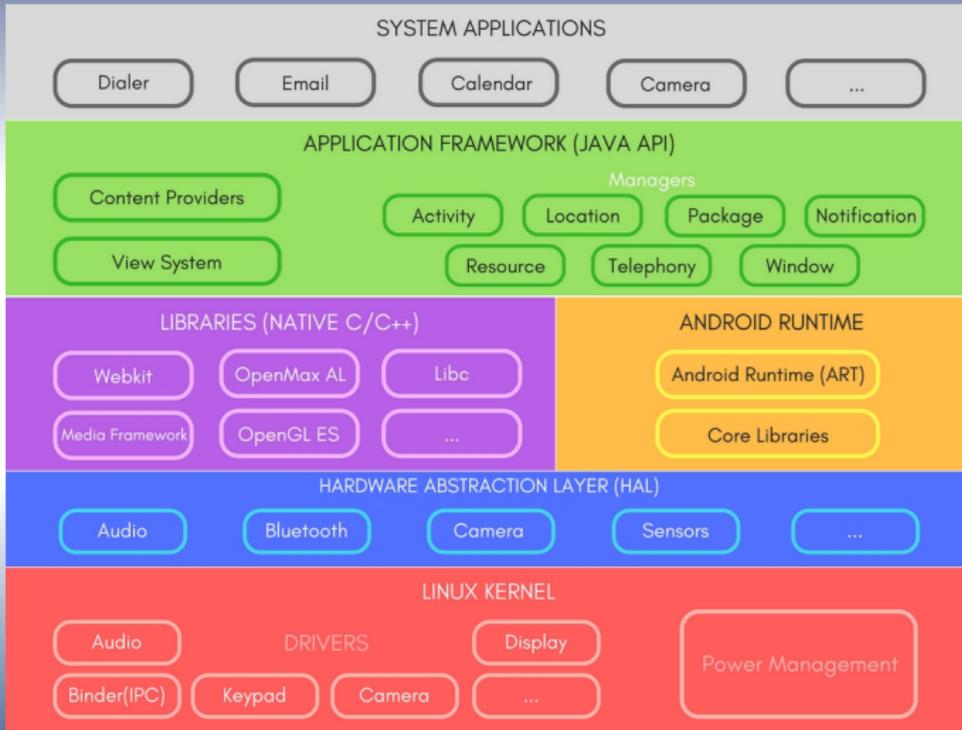# Nombre d'utilisateurs des différentes versions (accès au Play Store)

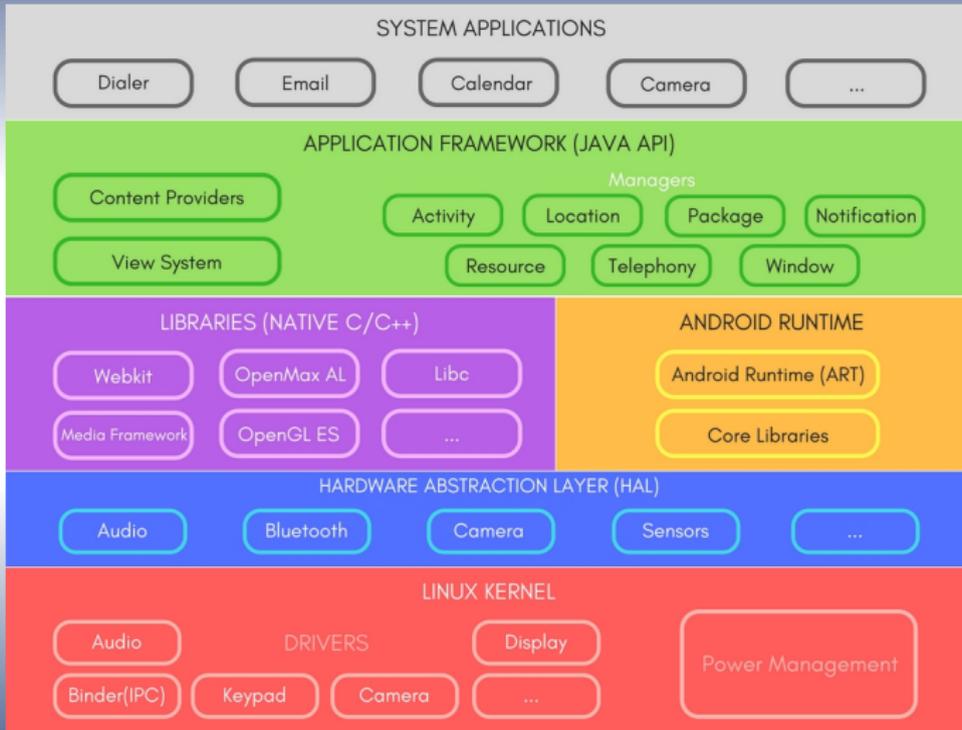| ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,9%
4.3 Jelly Bean	18	99,7%
4.4 KitKat	19	99,7%
5.0 Lollipop	21	98,8%
5.1 Lollipop	22	98,4%
6.0 Marshmallow	23	96,2%
7.0 Nougat	24	92,7%
7.1 Nougat	25	90,4%
8.0 Oreo	26	88,2%
8.1 Oreo	27	85,2%
9.0 Pie	28	77,3%
10. Q	29	62,8%
11. R	30	40,5%
12. S	31	13,5%

Marshmallow

Security

Fingerprint Authentication
Confirm Credential

System

App Linking
Adoptable Storage Devices

Multimedia

4K Display Mode
Support for MIDI
Create digital audio capture and playback objects
APIs to associate audio and input devices
List of all audio devices
Updated video processing APIs
Flashlight API
Reprocessing Camera2 API
Updated ImageWriter objects and Image Reader class

User Input

Voice Interactions
Assist API
Bluetooth Stylus Support

User Interface

Themeable ColorStateLists

Wireless & Connectivity

Hotspot 2.0
Improved Bluetooth Low Energy Scanning

Android for Work

Controls for Corporate-Owned, Single-Use devices
Silent install and uninstall of apps by Device Owner
Silent enterprise certificate access
Auto-acceptance of system updates
Delegated certificate installation
Data usage tracking
Runtime permission management
Work status notification

Last updated: August 4th, 2022

https://developer.android.com/about/versions/marshmallow/android-6.0.html

OK   Cancel

▸ Statistiques

Architecture globale

SYSTEM APPLICATIONS

Dialer | Email | Calendar | Camera | ...

APPLICATION FRAMEWORK (JAVA API)

Content Providers

Managers

Activity | Location | Package | Notification

View System

Resource | Telephony | Window

LIBRARIES (NATIVE C/C++)

Webkit | OpenMax AL | Libc

Media Framework | OpenGL ES | ...

ANDROID RUNTIME

Android Runtime (ART)

Core Libraries

HARDWARE ABSTRACTION LAYER (HAL)

Audio | Bluetooth | Camera | Sensors | ...

LINUX KERNEL

Audio | DRIVERS | Display

Binder(IPC) | Keypad | Camera | ...

Power Management

Libraries

Librairies C/C++ libres $\Rightarrow$

- **Surface Manager** : affichage 2D/3D
- **Media Framework** : codecs basés sur librairie OpenCORE
- **SQLite** : moteur de base de données relationelle
- **OpenGL—ES** : implémentation OpenGL pour l'embarqué
- **FreeType** : rendu bitmap de polices vectorielles
- **WebKit** : moteur de rendu de pages Web
- **SGL** : moteur graphique 2D
- **SSL** : communications réseaux sécurisées
- **libc** : dérivé de la librairie C (BSD) pour l'embarqué (Bionic libc)

Architecture : Application Framework

SYSTEM APPLICATIONS

| Dialer | Email | Calendar | Camera | ... |

APPLICATION FRAMEWORK (JAVA API)

Managers

Content Providers

Activity | Location | Package | Notification

View System

Resource | Telephony | Window

LIBRARIES (NATIVE C/C++)

Webkit | OpenMax AL | Libc

Media Framework | OpenGL ES | ...

ANDROID RUNTIME

Android Runtime (ART)

Core Libraries

HARDWARE ABSTRACTION LAYER (HAL)

Audio | Bluetooth | Camera | Sensors | ...

LINUX KERNEL

Audio | DRIVERS | Display

Binder(IPC) | Keypad | Camera | ...

Power Management

Application framework 1/2

Modules $\Rightarrow$

- **Content Providers** : gestion de l'accès aux données (appli et autres applis)
- **View System** : gestion des composants graphiques et des événements utilisateurs liés
- **Activity Manager** : cycle de vie et navigation entre applications
- **Location Manager** : accès aux fonctions de localisation

Application framework 2/2

Modules $\Rightarrow$

- **Package Manager** : accès aux API tierces installées

- **Notification Manager** : gestion du système de notifications

- **Resource Manager** : gestion des ressources (images, définition des layouts, etc.)

- **Telephony Manager** : accès aux services de téléphonie

- **Window Manager** : gestion GUI (z-ordered, mise en page, etc.)

Architecture : Android Runtime
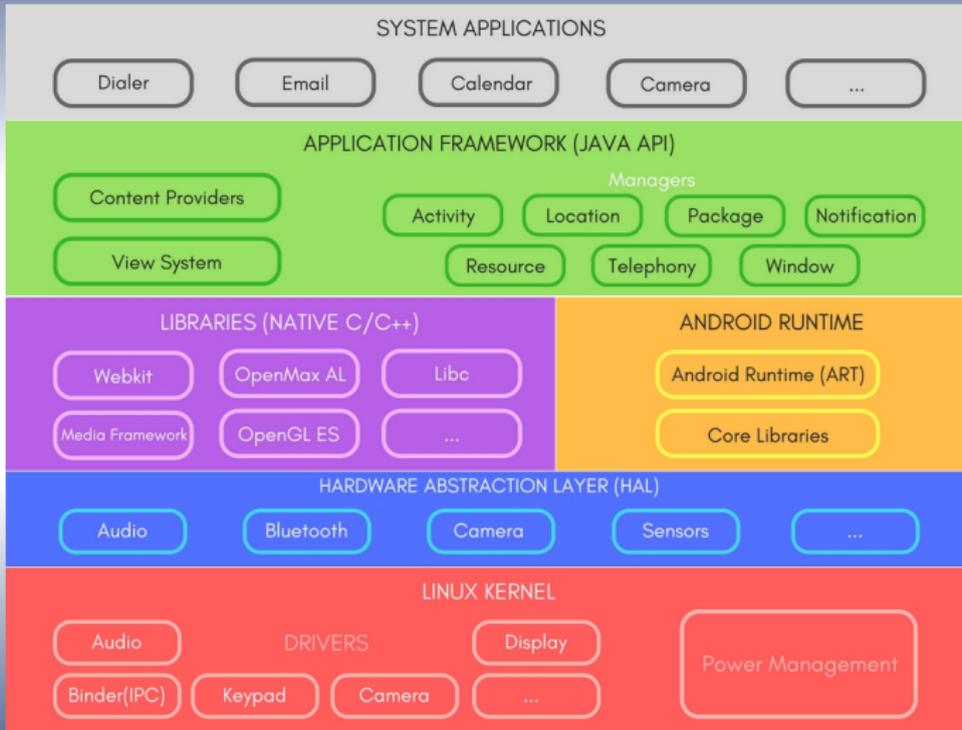
Construction d'un APK

APK : Android package, archive contenant les ressources nécessaires à l'exécution d'une application Android
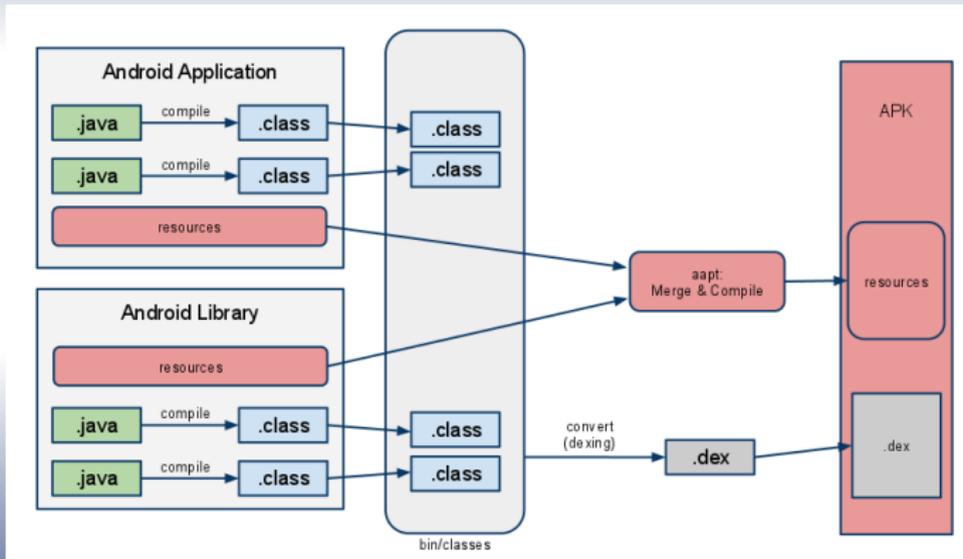
Plan

Prérequis pour créer une application Android :

Connaissances Techniques

- Android (runtime + framework) $\Rightarrow$ **POO + Java** / Kotlin

Android *Command line tools*

- Android **SDK Manager**
- **Android Virtual Device** Manager (AVD)
- **Android Debug Bridge** (adb) . . .

Un IDE

- **Android Studio** ⟨ ▸ téléchargement ⟩ / Eclipse

Command line tools (archive zip)

Décompression dans un répertoire, *e.g.* Android-sdk.
Répertoire référencé **SDK_PATH** dans la suite
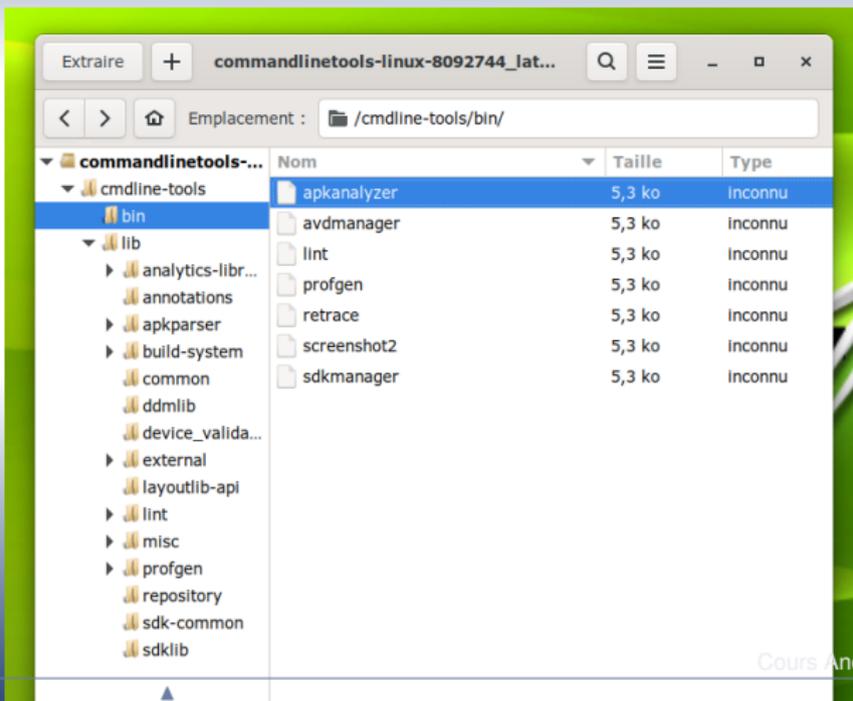
SDK Manager
Onglet SDK tools

SDK Manager
Onglet SDK platform : APIs Android

Android Virtual Device Manager (AVD)

Outils requis

IDE pour programmer une application Android :

2 solutions :

- Eclipse avec le plugin **Andmore** :
  - taper android dans le marketplace d'Eclipse

- **Android Studio** (supporté par Google) :
  - https://developer.android.com/studio

Android Studio

Contenu de l'archive (téléchargera tout le reste : tools, sdk, etc.)
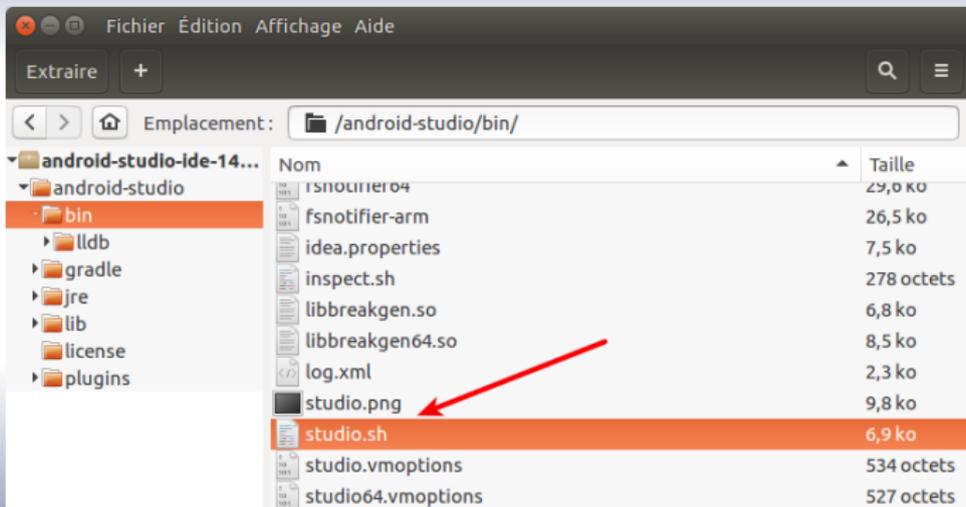
Android Studio
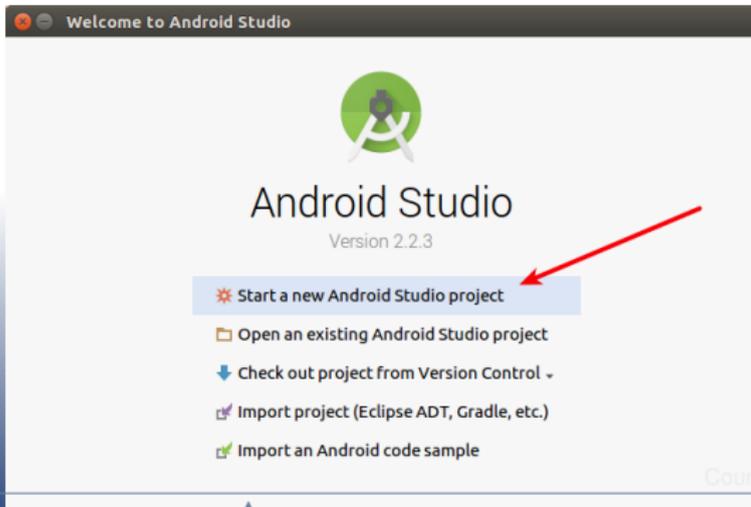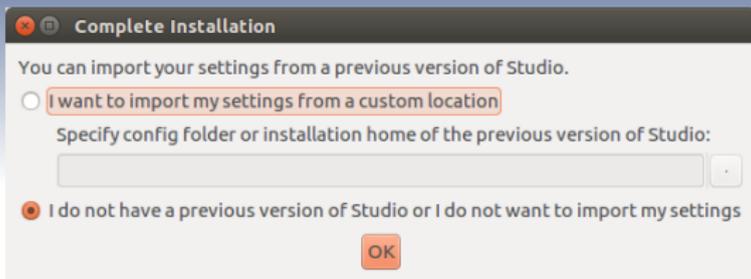
Complete Installation

You can import your settings from a previous version of Studio.

○ I want to import my settings from a custom location

Specify config folder or installation home of the previous version of Studio:

● I do not have a previous version of Studio or I do not want to import my settings

OK

Welcome to Android Studio

Android Studio

Version 2.2.3

⚙ Start a new Android Studio project

📁 Open an existing Android Studio project

⬇ Check out project from Version Control ▾

📝 Import project (Eclipse ADT, Gradle, etc.)

📝 Import an Android code sample

Plan

New project 1
Choix du squelette de l'activité principale
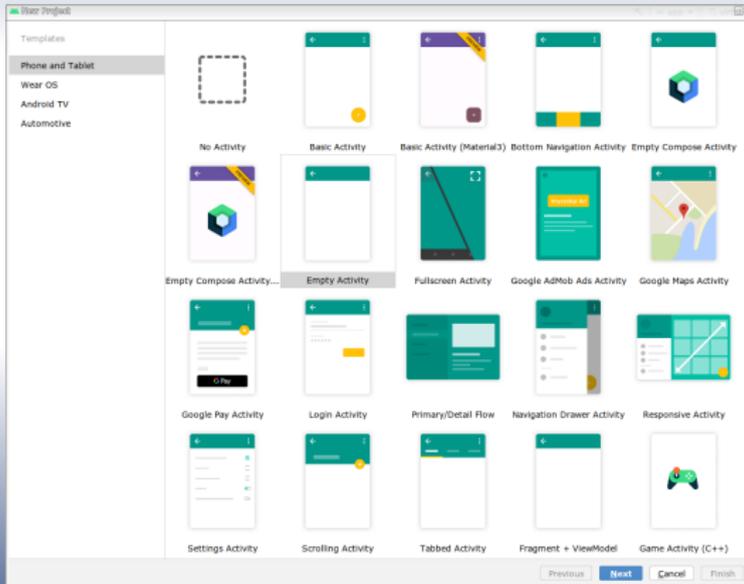
New project 2
Caractéristiques principales

New project 3
Sélection de l'API Minimum

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,9%
4.3 Jelly Bean	18	99,7%
4.4 KitKat	19	99,7%
5.0 Lollipop	21	98,8%
5.1 Lollipop	22	98,4%
6.0 Marshmallow	23	96,2%
7.0 Nougat	24	92,7%
7.1 Nougat	25	90,4%
8.0 Oreo	26	88,2%
8.1 Oreo	27	85,2%
9.0 Pie	28	77,3%
10. Q	29	62,8%
11. R	30	40,5%
12. S	31	13,5%

Lollipop

User Interface
Material design support
Concurrent documents and activities in the recents screen
WebView updates
Screen capturing and sharing

Notifications
Lock screen notifications
Notifications metadata

Graphics
Support for OpenGL ES 3.1
Android Extension Pack

Media
Camera API for advanced camera capabilities
Audio playback
Media playback control
Media browsing

Storage
Directory selection

Wireless & Connectivity
Multiple network connections
Bluetooth Low Energy
NFC enhancements

Battery - Project Volta
Scheduling jobs
Developer tools for battery usage

https://developer.android.com/about/versions/android-5.0.html

Printing Framework
Render PDF as bitmap

System
App usage statistics

Testing & Accessibility
Testing and accessibility improvements

IME
Easier switching between input languages

Manifest Declarations
Declarable required features
User permissions

Last updated: August 4th, 2022

OK   Cancel

F. Michel

Paramétrage des noms de fichiers
Ce choix n'est plus possible…
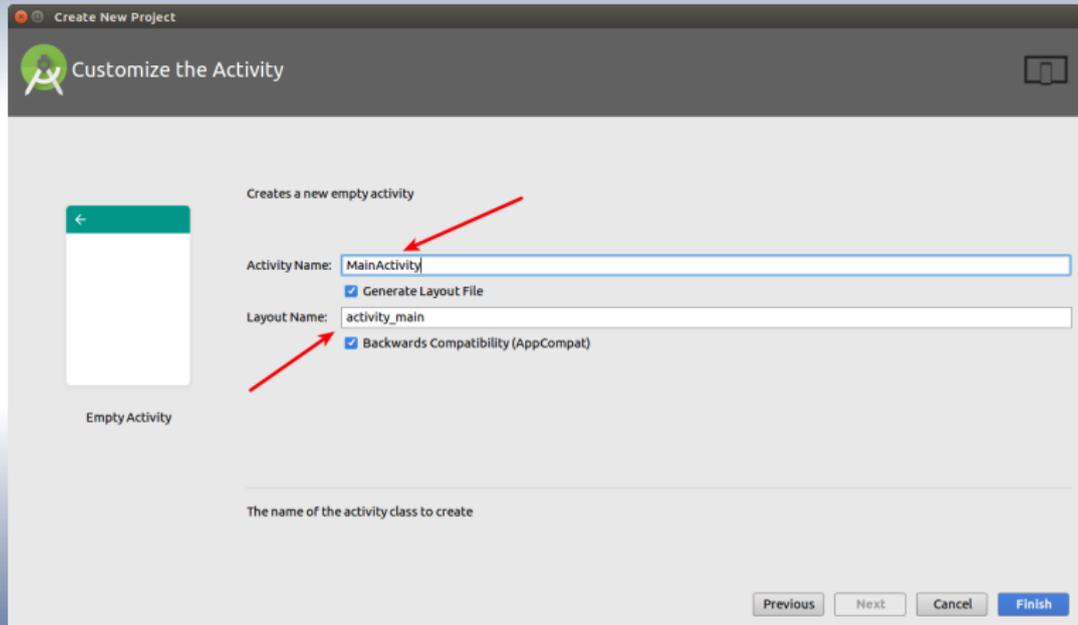
Téléchargement de l'API si nécessaire

Vue globale dans l'IDE
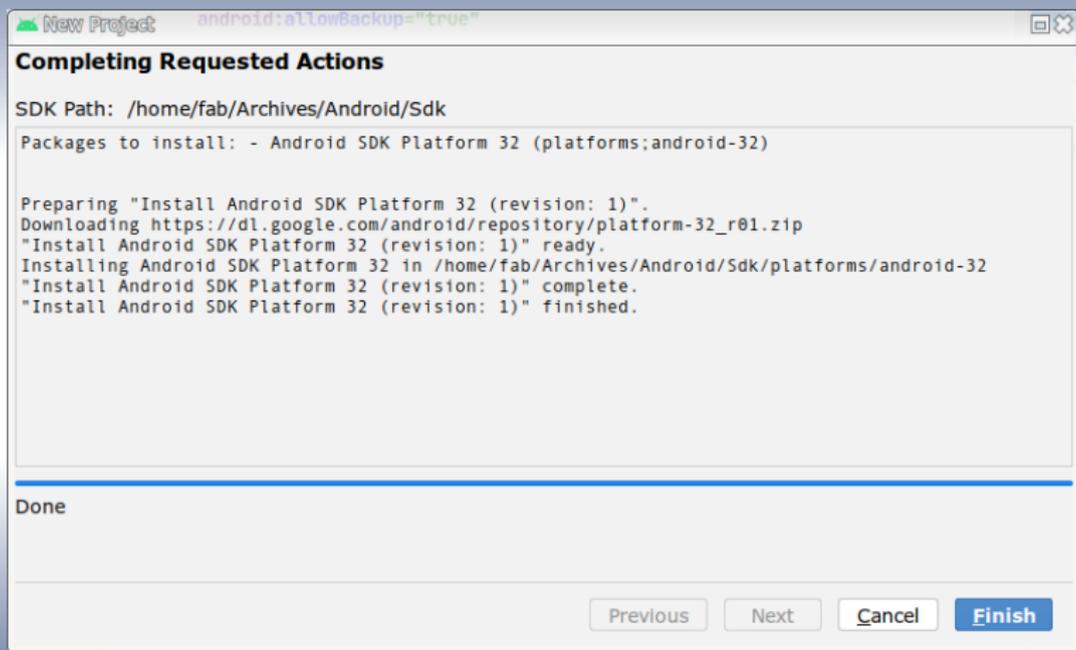
Contenu du projet

Dossier App

- **build**/ : fichier générés par le projet
- **libs**/ : librairies additionnelles ("à la main")
- **src**/ : intégralité des sources pour le développement (appli + tests)
- **build.gradle** : définition des options du build
- **proguard-rules.pro** : règles additionnelles pour le build
- **.gitignore** : définition des fichiers ignorés par git

Contenu du projet

Dossier App/src/

- **androidTest/** : tests de l'application dans l'environnement android

- **test/** : tests internes à l'application

- **main** : sources de l'application

  - **main/java** : sources Java, e.g. la classe qui lance l'activité
  - **main/res** : les ressources de l'application

    - **drawable-(h)(m)(l)dpi/** images dans différentes résolutions
    - **layout/** GUI design général
    - **values/** valeurs des variables
    - **menu/** définition des menus

- **main/AndroidManifest.xml** : description et éléments-clés de l'application (nom, activité principale, intents, etc.)

app/src/main/AndroidManifest.xml : description

Manifest complet

Plan

res/layout/activity_main.xml
disposition de la GUI

activity_main.xml ⊠

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.examplefr.iutmontp.helloworld.MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

Éditeur de layout (*text mode*)
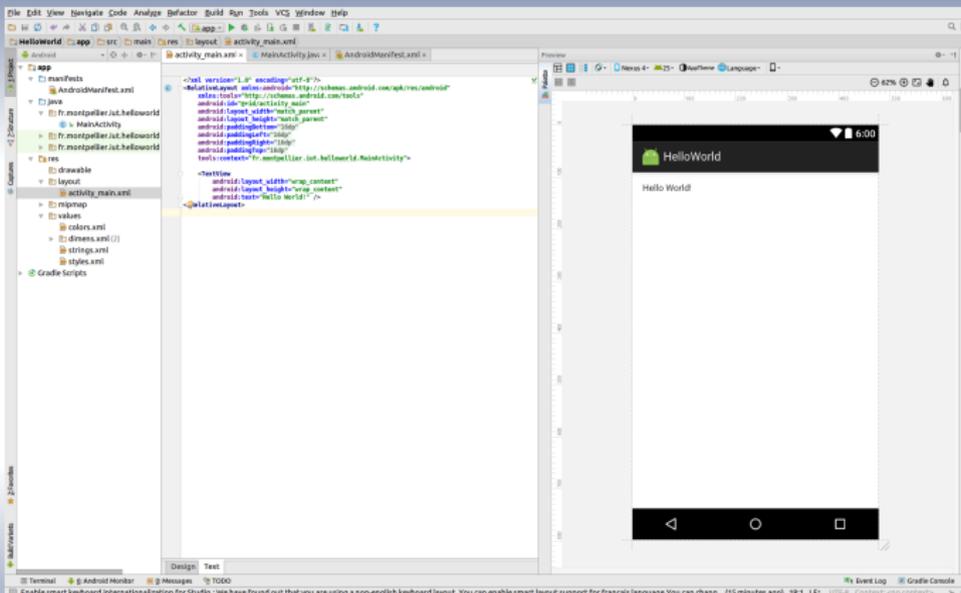
Éditeur de layout (*design mode*)
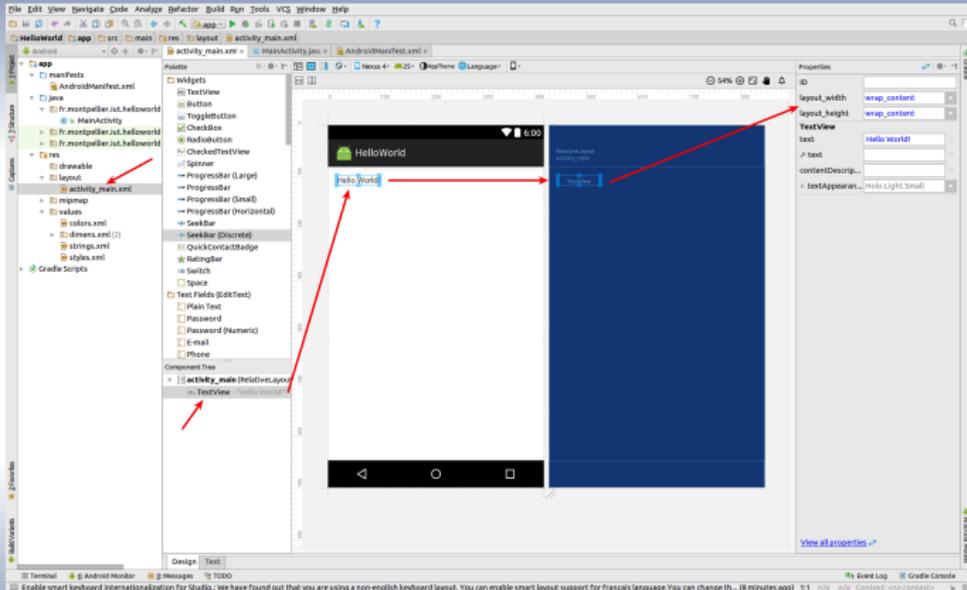
strings.xml : valeurs des variables

Éditeur pour *strings.xml*

/src/.../MainActivity.java : activité
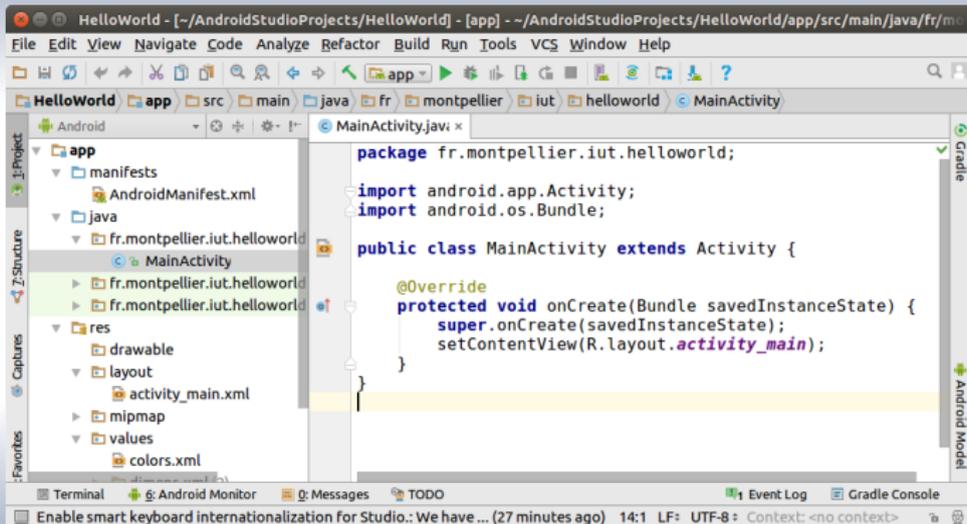
/build/. . ./R.java : définition d'identifiants


```
Fichier  Édition  Affichage  Rechercher  Outils  Documents  Aide

Ouvrir    ▼    ⏏                                          Enregistrer

/* AUTO-GENERATED FILE.  DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found.  It
 * should not be modified by hand.
 */

package fr.montpellier.iut.helloworld;

public final class R {
    public static final class attr {
    }
    public static final class color {
        public static final int colorAccent=0x7f050000;
        public static final int colorPrimary=0x7f050001;
        public static final int colorPrimaryDark=0x7f050002;
    }
    public static final class dimen {
        public static final int activity_horizontal_margin=0x7f040000;
        public static final int activity_vertical_margin=0x7f040001;
    }
    public static final class id {
        public static final int activity_main=0x7f080000;
    }
    public static final class layout {
```

Plan

Exécution sur un vrai dispositif

Mise en place du dispositif pour le développement

- connexion usb : ‣ drivers windows ‣ configuration linux

- sous linux, la commande *lsusb* liste les connexions usb
  ‣ plus d'information

- activation *USB debugging* sur le dispositif :
  à partir de Android 4.2 : Settings → About phone (7 tapes
  successives).

Exécution du projet

Une fois déployée

Résultat →
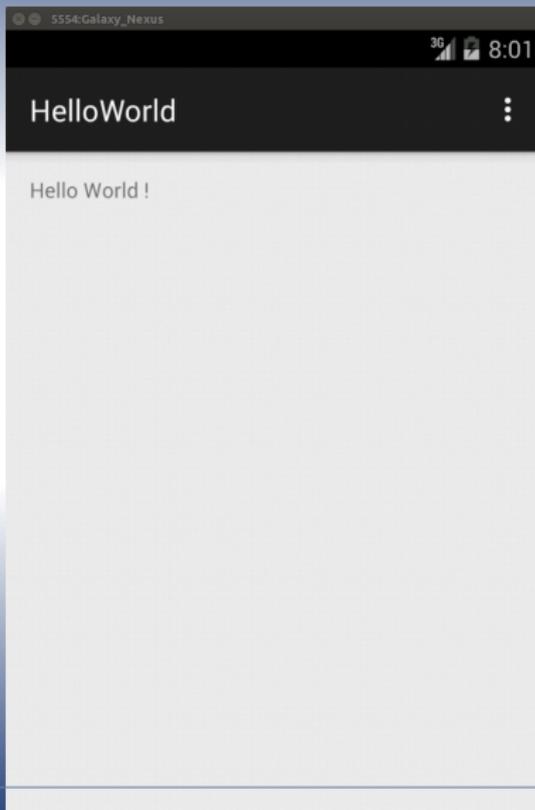
Exécution sur un émulateur avec AVD
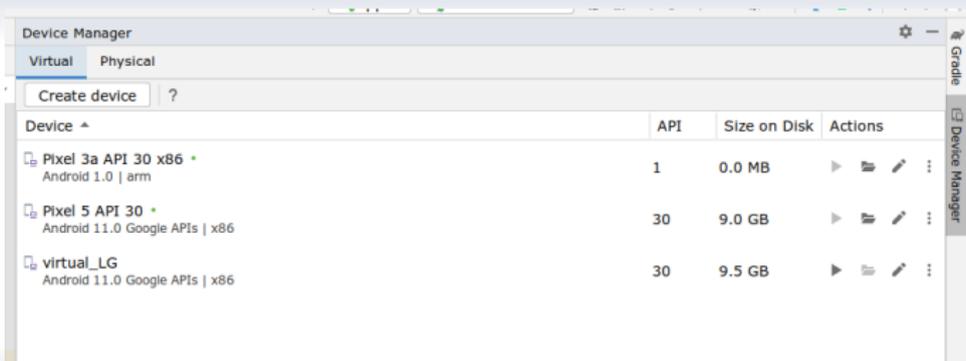
Création d'une device virtuelle
Choix du modèle

Exécution du projet

Choix de l'image système
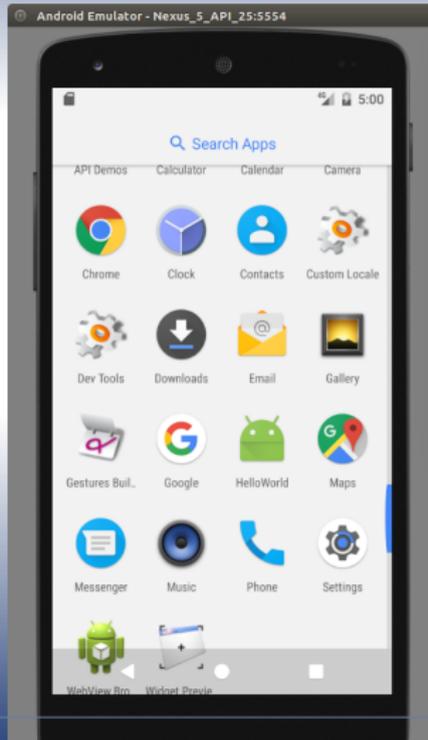
Paramétrage avancé de la device

Start →

Exécution du projet sur l'AVD

L'application est maintenant disponible sur la cible (virtuelle ou non)

Résumé global

- La technologie Android :
  - Noyau Linux
  - Programmation **Java** / Kotlin

- Programmation Android
  - des outils pour le développement : SDK manager / AVD / un IDE.
  - une structure de projet standardisée (manifest, src, res, etc.)
  - des dispositifs pour le déploiement (virtuels et réels)